

(19)日本国特許庁 (JP)

(12) 公開特許公報 (A)

(11)特許出願公開番号

特開2002-116838

(P2002-116838A)

(43)公開日 平成14年4月19日(2002.4.19)

(51)Int.Cl. ⁷	識別記号	F I	テマコード*(参考)
G 0 6 F 1/00 9/445		G 0 6 F 12/00	5 1 0 B 5 B 0 1 7
12/00	5 1 0	12/14	5 3 7 H 5 B 0 7 6
	5 3 7	13/00	3 2 0 B 5 B 0 8 2
12/14	3 2 0	15/00	5 3 0 B 5 B 0 8 5
			3 3 0 Z 5 J 1 0 4
審査請求 有		請求項の数19 O L (全 11 頁)	最終頁に統く

(21)出願番号 特願2001-195753(P2001-195753)

(71)出願人 390009531

(22)出願日 平成13年6月28日(2001.6.28)

インターナショナル・ビジネス・マシーンズ・コーポレーション

(31)優先権主張番号 0 0 1 1 3 8 5 7. 7

INTERNATIONAL BUSIN
ESS MASCHINES CORPO
RATION

(32)優先日 平成12年6月30日(2000.6.30)

アメリカ合衆国10504、ニューヨーク州
アーモンク (番地なし)

(33)優先権主張国 欧州特許庁 (EP)

(74)代理人 100086243

弁理士 坂口 博 (外2名)

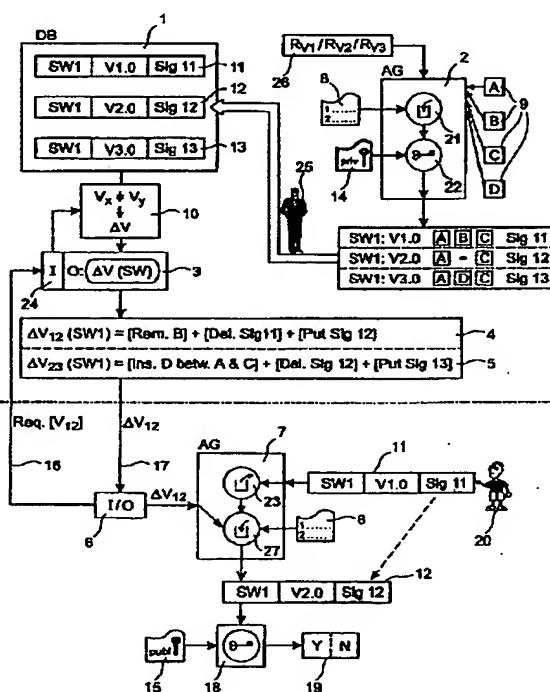
最終頁に統く

(54)【発明の名称】 コードを更新するためのデバイスおよび方法

(57)【要約】 (修正有)

【課題】 伝送するデータの量を低減するが、機能性における変更を容易にし、Javaコード・コンテナなど、署名付きコードのセキュリティ特性を保持する方法を提供する。

【解決手段】 ソフトウェア取得エンティティ20が既存の第1の署名付きコード片11から第2の署名付きコード片12に達することができるようにする、ソフトウェア・プロバイダのための方法を対象とする。両方のコード片は、ソフトウェア・プロバイダ側で、第1のソフトウェア・アーカイブ生成器2を使用することによって、生成命令の使用下で生成された。ソフトウェア・プロバイダは、第1の署名付きコード片11から第2の署名付きコード片12に達するために必要なステップを含む差分コード4をソフトウェア取得エンティティ20に提供する。



1

2

【特許請求の範囲】

【請求項1】ソフトウェア取得エンティティ（20）が既存の第1の署名付きコード片（11）から第2の署名付きコード片（12、13）に達することができるようする、ソフトウェア・プロバイダ（25）のための方法であって、両方のコード片（11、12、13）は、第1のソフトウェア・アーカイブ生成器（2）を使用することによって、生成命令（8）の使用下で生成されており、前記第1の署名付きコード片（11）から前記第2の署名付きコード片（12、13）に達するために必要なステップを含む差分コード（4、5）を前記ソフトウェア取得エンティティ（20）に提供するステップを含み、この差分コード（4、5）は、前記ソフトウェア取得エンティティ（20）側で、第2のソフトウェア・アーカイブ生成器（7）によって前記第1の署名付きコード片（11）と結合されて前記第2の署名付きコード片（12、13）を生成するように使用可能であり、それにより、前記第1のソフトウェア・アーカイブ生成器（2）によって両方のコード片（11、12、13）の前記生成のために使用されたそれらの生成命令（8）が前記第2のソフトウェア・アーカイブ生成器（7）に供給される方法。

【請求項2】前記生成命令（8）が、前記ソフトウェア・プロバイダ（25）によって、好ましくは前記第2のソフトウェア・アーカイブ生成器（7）と共に前記ソフトウェア取得エンティティ（20）に提供される、請求項1に記載の方法。

【請求項3】前記コード片（11、12、13）が、秘密鍵（14）を使用して署名される、請求項1または2に記載の方法。

【請求項4】前記署名付きコード片（11、12、13）が、前記ソフトウェア・プロバイダ（25）側の記憶装置（1）に格納される、請求項1ないし3のいずれか一項に記載の方法。

【請求項5】前記差分コード（4、5）が、好ましくは前記第1のソフトウェア・アーカイブ生成器（2）によって作成され、前記第1のソフトウェア・アーカイブ生成器（2）が前記第2の署名付きコード片（12、13）を生成する、請求項1ないし4のいずれか一項に記載の方法。

【請求項6】3つ以上のコード片（11、12、13）が格納される場合、前記差分コード（4、5）が前記コード片（11、12、13）の部分集合の間でのみ生成される、請求項1ないし5のいずれか一項に記載の方法。

【請求項7】前記第1のコード片（11）から前記第2のコード片（13）に達するために、いくつかの差分コード（4、5）が必要とされ、これらの差分コード（4、5）が单一の差分コードに併合されて前記ソフトウェア取得エンティティ（20）に提供される、請求項

6に記載の方法。

【請求項8】前記第1および第2のコード片（11、12、13）が、前記ソフトウェア・プロバイダ（25）側で、前記ソフトウェア取得エンティティ（20）から受信された要求（16）から対応する識別子を導出することによって識別される、請求項1ないし7のいずれか一項に記載の方法。

【請求項9】既存の第1の署名付きコード片（11）から第2の署名付きコード片（12、13）に達するソフトウェア取得エンティティ（20）のための方法であって、両方のコード片（11、12、13）は、ソフトウェア・プロバイダ（25）側で、第1のソフトウェア・アーカイブ生成器（2）を使用することによって、生成命令（8）の使用下で生成されており、

前記第1の署名付きコード片（11）から前記第2の署名付きコード片（12、13）に達するために必要なステップを含む差分コード（4、5）の送達のために、コード修正要求（16）を前記ソフトウェア・プロバイダ（25）に送信するステップと、

前記差分コード（4、5）を受信するステップと、第2のソフトウェア・アーカイブ生成器（7）を使用することによって前記差分コード（4、5）を前記第1の署名付きコード片（11）と結合し、それにより前記第2の署名付きコード片（12、13）を生成するステップとを含み、それにより、前記第1のソフトウェア・アーカイブ生成器（2）によって両方のコード片（11、12、13）の前記生成のために使用されたそれらの生成命令（8）が前記第2のソフトウェア・アーカイブ生成器（7）に供給される方法。

【請求項10】前記生成命令（8）が前記ソフトウェア・プロバイダ（25）から、好ましくは前記第2のソフトウェア・アーカイブ生成器（7）と共に受信される、請求項9に記載の方法。

【請求項11】前記コード片（11、12、13）が、秘密鍵（14）を使用することによって署名され、署名（Sig11、Sig12、Sig13）が、対応する公開鍵（15）を使用することによって検証可能である、請求項9または10に記載の方法。

【請求項12】前記第1および第2のコード片（11、12、13）が、前記コード修正要求（16）において対応する識別子を与えることによって前記ソフトウェア取得エンティティ（20）によって識別される、請求項9ないし11のいずれか一項に記載の方法。

【請求項13】請求項1ないし12のいずれか一項に記載の方法を実行するためのプログラム・コード手段を含む、コンピュータ・プログラム製品。

【請求項14】コンピュータ可読媒体上に格納された前記プログラム・コード手段を含む、請求項13に記載のコンピュータ・プログラム製品。

【請求項15】ソフトウェア取得エンティティ（20）

が、既存の第1の署名付きコード片(11)から第2の署名付きコード片(12、13)に達することができるようとするコード修正イネーブラであって、両方のコード片(11、12、13)は、第1のソフトウェア・アーカイブ生成器(2)を使用することによって、生成命令(8)の使用下で生成されており、前記第1の署名付きコード片(11)から前記第2の署名付きコード片(12、13)に達するために必要なステップを含む差分コード(4、5)を生成するための差分コード生成器(10)を含み、この差分コード(4、5)は、前記ソフトウェア取得エンティティ(20)側で、第2のソフトウェア・アーカイブ生成器(7)によって前記第1の署名付きコード片(11)と結合されて前記第2の署名付きコード片(12、13)を生成するように使用可能であり、それにより、前記生成命令(8)が前記第2のソフトウェア・アーカイブ生成器(7)に供給され、さらに、

前記ソフトウェア取得エンティティ(20)に前記差分コード(4、5)を提供するための出力ユニット(3)を含む、コード修正イネーブラ。

【請求項16】前記差分コード(4、5)の送達のためのコード修正要求(16)を前記ソフトウェア取得エンティティ(20)から受信するための入力ユニット(24)をさらに含む、請求項15に記載のコード修正イネーブラ。

【請求項17】前記コード片(11、12、13)を生成命令(8)の使用下で生成するための第1のソフトウェア・アーカイブ生成器(2)をさらに含む、請求項15または16に記載のコード修正イネーブラ。

【請求項18】既存の第1の署名付きコード片(11)から第2の署名付きコード片(12、13)に達するためのコード修正デバイスであって、両方のコード片(11、12、13)は、ソフトウェア・プロバイダ(25)側で、第1のソフトウェア・アーカイブ生成器(2)を使用することによって、生成命令(8)の使用下で生成されており、

受信された差分コード(4、5)を前記第1の署名付きコード片(11)と結合し、それにより前記第2の署名付きコード片(12、13)を生成するための第2のソフトウェア・アーカイブ生成器(7)を含み、それにより、前記第1のソフトウェア・アーカイブ生成器(2)によって両方のコード片(11、12、13)の前記生成のために使用されたそれらの生成命令(8)が前記第2のソフトウェア・アーカイブ生成器(7)に供給される、コード修正デバイス。

【請求項19】コード修正要求(16)を前記ソフトウェア・プロバイダ(25)に送信し、前記差分コード(4、5)を受信するための入出力ユニット(6)をさらに含む、請求項18に記載のコード修正デバイス。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】本発明は、ソフトウェア取得エンティティが既存の第1の署名付きコード片から第2の署名付きコード片に達することができるようとする、ソフトウェア・プロバイダのための方法に関する。本発明は、既存の第1の署名付きコード片から第2の署名付きコード片に達するソフトウェア取得エンティティのための方法にも関する。

【0002】

10 【従来の技術】今日の典型的なJavaアプリケーションは、非常に大きく、数千個ものクラスからなる傾向がある。さらに、Javaコードを実行する、普及しているウェブブラウザでは、異なるJavaパッケージングおよび署名フォーマットが必要なので、コードが通常2度配布され、それによりさらにアプリケーションのサイズが増大する。このように複雑さが増すことは、より多くのプログラミングのエラーが、プログラムがすでに顧客へロール・アウトされたときになって初めて明らかになることに通じる。これにより、誤り訂正を顧客に送ることが必要となる。Javaプログラムは、暗号化署名付きコンテナにおいて出荷されるので、これまで、完全なJavaコンテナは、場合によっては数千個ものクラスを含み、その中のただ1つのクラス・ファイルが変更されたときでさえ再配布する必要がある。

【0003】

【発明が解決しようとする課題】独立クレームによる本発明の目的は、伝送するデータの量を低減するが、機能性における変更を容易にし、Javaコード・コンテナなど、署名付きコードのセキュリティ特性を保持する方法を提供することである。

【0004】

【課題を解決するための手段】したがって、本明細書で解決される主な問題は、たとえば、コードにおけるエラーを修正する、あるいはコードを新しい機能セットへアップグレードするなど、第1のコード片を修正して第2のコード片に達するときに、顧客に出荷されるデータの量の低減である。

【0005】よって、本発明は、ソフトウェア取得エンティティが既存の第1の署名付きコード片から第2の署名付きコード片に達することができるようとする、ソフトウェア・プロバイダのための方法を対象とする。両方のコード片は、ソフトウェア・プロバイダ側で、第1のソフトウェア・アーカイブ生成器を使用することによって、生成命令の使用下で生成された。ソフトウェア・プロバイダは、第1の署名付きコード片から第2の署名付きコード片に達するために必要なステップを含む差分コードをソフトウェア取得エンティティに提供する。差分コードは、ソフトウェア取得エンティティ側で、第2のソフトウェア・アーカイブ生成器によって第1の署名付きコード片と結合可能であり、第2の署名付きコード片

を生成する。したがって、第1のソフトウェア・アーカイブ生成器によって両方のコード片の生成のために使用されたそれらの生成命令が第2のソフトウェア・アーカイブ生成器に供給される。

【0006】第1のソフトウェア・コンポーネント併合ユニットは、第1の署名付きコード片を生成するための生成命令を使用する。ユーザ側でそれらの生成命令を使用して第2の署名付きコード片を生成するので、生成命令がソフトウェア取得エンティティへ、ソフトウェア・プロバイダによって、好ましくは第2のソフトウェア・アーカイブ・ジェネレータと共に提供される場合に有利である。それにより、ユーザが、第2の署名付きコード片の正しい生成を許可する1組のツールを所有しているという、よりよい保証が存在する。

【0007】ソフトウェア・プロバイダ側のシステムがさらに、秘密鍵へのアクセスを有する署名ユニットを含むことができる。コード片は、秘密鍵を使用して署名される。公開／秘密鍵システムは、非常に広く行き渡り、よく知られている暗号化システムであり、したがって実施しやすく使いやすい。

【0008】差分コードが、好ましくは第1のソフトウェア・アーカイブ生成器によって作成され、第1のソフトウェア・アーカイブ生成器が第2の署名付きコード片を生成する場合、有利である。差分コードが、第2のコード片に達する方法のステップをリフェルセット(referrals)するので、差分ファイルに入れられる情報が自動的に作成され、生成処理中に使用可能である。

【0009】既存の第1の署名付きコード片から第2の署名付きコード片に達するソフトウェア取得エンティティのための方法であって、それにより両方のコード片が、ソフトウェア・プロバイダ側で、第1のソフトウェア・アーカイブ生成器を使用することによって、生成命令の使用下で生成されており、この方法は、第1の署名付きコード片から第2の署名付きコード片に達するために必要なステップを含む差分コードの送達のために、コード修正要求をソフトウェア・プロバイダに送信するステップと、差分コードを受信し、第2のソフトウェア・アーカイブ生成器を使用することによって差分コードを第1の署名付きコード片に結合し、それにより第2の署名付きコード片を生成するステップとを含む。第1のソフトウェア・アーカイブ生成器によって両方のコード片の生成のために使用されたそれらの生成命令が第2のソフトウェア・アーカイブ生成器に供給される。ユーザには、完全に新しい第2の署名付きコードではなく、小さい差分コードのみを使用する利点がある。差分コードを、ネットワークを介してダウンロードすることで、著しく少ない時間しかかかる可能性がある。ソフトウェア・プロバイダにとっても、このことが当たる。差分ファイルを提供するためのコストも、完全バージョンの第2の署名付きコード片のコストよりも低くなるで

あろう。

【0010】提案した方法は、提案した方法を実行するためのプログラム・コードを含むコンピュータ・プログラム製品の形式で実現し、市場に出すことができる。このコンピュータ・プログラム製品を、コンピュータ可読媒体上に格納することができる。

【0011】本発明を理解するため、まず、既存の状況についてより詳細に説明する。普通または正当な人物など、あるエンティティが特定のコード片を取得しており、この人物を以下でユーザとも呼び、この人物がこのコードをコード片の異なる片で置き換えることを望むと仮定し、両方のコードをソフトウェアとも呼び、先に取得されたソフトウェア片の更新されたバージョンであることが好ましい。以下では、具体的な例を挙げるために、ユーザが、自分がすでに取得したソフトウェアの更新を有することを望むと仮定する。典型的には、ユーザが、すでに取得したソフトウェア片の作成元に更新を要求する。次いで、ソフトウェアの作成者が、更新された完全バージョンのソフトウェア、あるいは、まだ更新されていないソフトウェアの存在において実行されているときに、次いで既存のソフトウェアを更新されたソフトウェアに変更するソフトウェアを、ユーザに送信することができる。署名付きソフトウェアの場合、すなわち、ソフトウェアの完全性およびオリジナリティを、秘密／公開鍵暗号化方式を使用して検証することができる場合、このような更新では今まで、ソフトウェアの作成者が完全バージョンの更新を送信する必要があり、これは、先に取得されたソフトウェアを更新と共に組立てることにより、実際にすべての場合において、検証ツールによってオリジナルでないと認識されるであろう異なる署名に通じるからである。しかし、これは、ユーザが署名付きソフトウェアの署名を使用して、オリジナル、すなわち、無許可の手によって触れられていないソフトウェアを、ある無許可のエンティティによって修正されたソフトウェアから区別できるようにするためである。ユーザはこの署名の利点を、更新が行われるときにも持続することを望む。ユーザはクライアントとも呼ばれ、ソフトウェア・プロバイダはサーバとも呼ばれる。これは、典型的な実現では、ソフトウェア・プロバイダが自分で行うのではなく、クライアント・システムを動作するユーザに所望の更新コードを提供するために必要なステップを自動的に実行するサーバなど、コンピュータ・システムにおいて自動化された提供機能を有する可能性があるからである。ユーザ側では、更新要求に加えて最後の更新の実行を、クライアントと呼ばれる、コンピュータ化されたシステムにおいて自動化することができる。

【0012】更新されるコードのための暗号化署名の再作成を許可する、暗号により保護された差分ファイルを使用する解法を提案する。クライアント側ソフトウェア

の署名の再作成のために提示した手法は、いかなる新しい暗号化機能性をもクライアント側で使用する必要がないが、これをサーバ側、およびサードパーティ実行環境、たとえば、ウェブブラウザに制限するという利点を有する。この一般的な戦略において構築されたソフトウェアは、暗号化の方法を使用しないので、いかなるエクスポート制限も受けないという利点を有する。さらに、これは明らかにより安全であり、これは、署名鍵、すなわち、ソフトウェア・プロバイダの秘密鍵が、クライアント側で署名を再作成するために必要とされないからである。

【0013】本発明は、インターネットなど本質的に不確かな媒体上の分野において、すでにインストールされ、配備されている署名付きアプリケーションを、安全に更新するための新しい手法を使用する。これは、上記の目的をとても効率的な方法で達成可能にする様々な技術を包含する。本質において、この概念について、オープン・セキュリティ・プロトコル、および、JARまたはCABなど署名付き文書フォーマットに適用された識別技術と結合して、オンライン・バージョニング検査を使用するために説明することもできる。

【0014】

【発明の実施の形態】以下で、本発明の様々な例示的実施形態について説明する。

【0015】第1の実施形態

図1には、いくつかの署名付きコード片11、12、13を含むデータベースを含む記憶装置1が示されている。これらの署名付きコード片11、12、13は、第1のソフトウェア・アーカイブ生成器2によって生成されており、これもAGで示す。第1のソフトウェア・アーカイブ生成器2は、第1のソフトウェア・コンポーネント併合ユニット21を含み、これが生成命令8を使用する。

【0016】これはさらに署名ユニット22を含み、これが秘密鍵14へのアクセスを有する。図示の例では、4つのソフトウェア・コンポーネント9をA、B、C、Dで示し、これらが第1のソフトウェア・アーカイブ生成器2へ送り込まれ、これが生成命令8およびバージョン命令26に従って、これらのソフトウェア・コンポーネント9をアセンブリに組み立て、これをアーカイブまたはコンテナとも呼ぶ。

【0017】各コード片11、12、13について、これらのバージョン命令26が、何がその特定のコード片11、12、13に含まれるか、および、それがどのようにどこで含まれるかについて、コード特有の情報を含む。よって、ここでは、第1の署名付きコード片11について、これは規則RV1を含み、これは3つのコンポーネントA、B、Cが厳密にこの順序で入れられることを示す。第2の署名付きコード片12について、これは規則RV2を含み、これは2つのコンポーネントA、C

が厳密にこの順序で入れられることを示す。第3の署名付きコード片13について、これは規則RV3を含み、これは3つのコンポーネントA、D、Cが厳密にこの順序で入れられることを示す。生成命令8はより汎用の種類であり、いずれの署名付きコード片11、12、13の生成にも適切である命令を与える。次いで、このアセンブリが、秘密鍵14を使用して、署名ユニット22によって署名される。この生成は、ここでは3つの異なるバージョンに通じる。すなわち、第1のバージョンV1.0、第2のバージョンV2.0および第3のバージョンV3.0である。第1のバージョンV1.0は署名Sig11を有し、これにより第1の署名付きコード片11を構築する。第2のバージョンV2.0は署名Sig12を有し、これにより第2の署名付きコード片12を構築する。第3のバージョンV3.0は署名Sig13を有し、これにより第3の署名付きコード片13を構築する。この具体的な例では、第1のバージョンV1.0が3つのソフトウェア・コンポーネントA、B、Cおよび署名Sig11からなり、第2のバージョンV2.0が2つのソフトウェア・コンポーネントA、Cおよび署名Sig12のみからなり、第3のバージョンV3.0が3つのソフトウェア・コンポーネントA、D、Cおよび署名Sig13からなる。

【0018】上記で説明したように、これらの3つのコード片11、12、13が記憶装置1に格納される。記憶装置1が差分コード生成器10に接続され、これがその出力を出力ユニット3へ送達し、これが入力ユニット24に結合される。ここまででは、ソフトウェア・プロバイダ25側のインフラストラクチャについて説明した。

【0019】ソフトウェア取得エンティティ20は、ユーザ20とも呼ばれ、この側に第2のソフトウェア・アーカイブ生成器7が存在し、これはソフトウェア・コンポーネント・セパレータ23および第2のソフトウェア・コンポーネント併合ユニット27を含み、これが、ソフトウェア・プロバイダ側のものと等しい生成命令8の使用下で動作可能である。ユーザ20は第1の署名付きコード片11を有し、これを第2の署名付きコード片12に修正することを望む。したがって、ユーザ20が、どの更新が望まれるかについての情報を与える識別子が添付されるコード修正要求16を、ソフトウェア・プロバイダ25へ、入出力ユニット6を介して入力ユニット24に送信する。ここでは、識別子が、ユーザ20がソフトウェアSW1のバージョンV1.0からバージョンV2.0への更新を必要とする事を示す。識別子は、現在の機能セットを記述するバージョン番号にすることができ、いかなる文字列、または数値にすることもできる。

【0020】コード修正要求16が、ソフトウェア・プロバイダ25の入力ユニット24において受信され、その一式において、差分コード生成器10が2つのバージ

ヨンV1.0およびV2.0を比較し、対応する差分コードDV(SW1)を生成し、これは本明細書では第1の差分コード4、すなわちDV12(SW1)である。このコードの内容は、第1のバージョンV1.0を第2のバージョンV2.0へ修正するための命令であり、ソフトウェア・コンポーネントBが除去され、署名Sig11が除去されて署名Sig12で置き換えられることを示す。よって、差分コード生成器は、内容における差分、および2つの署名付きコンテナ・ファイル・ツール、すなわち第1の署名付きコード片11および第2の署名付きコード片12の署名を抽出するためのツールである。更新された署名付きのコンテナに含まれたすべてのエントリの厳密な順序が記録される。

【0021】よって、第1の差分コード4は、第1のコード片11から開始して第2のコード片12に達するために必要なすべてのステップを含む。これらのステップは、ユーザ側のソフトウェア・アーカイブ生成器7が、第2の署名付きコード片12を生成するために必要である精度と共に与えられ、これは、最後には、第2のコード片12に属する署名Sig12が、ユーザ側で更新されたコード片のための正しい署名であるという方法による。差分ファイル4がユーザ20のためのバージョン命令26を反映して、ユーザ20側で生成命令8のみが必要とされるようにする、ということができる。

【0022】特定のコード片の署名Sig11、Sig12、Sig13の検証が、コードの内部構造に依存する特定の結果を出し、これは典型的にはハッシュ技術が署名に使用されるからである。署名Sig11、Sig12、Sig13があるコード片について正しく識別され、検証されるために、よってこのコード片の内部構造が、署名されたコード片の内部構造と等しくなければならない。これは、秘密/公開鍵暗号システムが、特別に高い確率により、2つの異なるコード片についての結果が異なることを保証するからである。これは、典型的に使用される、固有の非対称に基づく暗号化方式のいずれにも当てはまり、暗号化されたファイルを生成するために使用される時間が、典型的には数桁分、暗号化されたファイルから暗号化されていないファイルに達するためには要するであろう時間よりもはるかに短い。後者の時間は、実際の場合に人間の寿命を越え、よってこれが暗号化の期間において安全として定義される。

【0023】よって、ユーザ側で第1のソフトウェア片11を第2のソフトウェア片12に修正するための処理が、結果の第2の署名付きコード片12が、元の第2の署名付きコード片12がソフトウェア・プロバイダ側で生成され署名されたときに有したものと厳密に同じ内部構造を有することを保証する方法で行われることが、保証されなければならない。第1の差分コード4が生成命令8およびコンポーネント併合ユニット21と共にツールボックスを形成し、これが、ユーザ側の生成処理がソ

フトウェア・プロバイダ側の生成処理と合致することを保証する。この合致により、ユーザ側で生成されるときの第2の署名付きコード片12の内部構造が、ソフトウェア・プロバイダ側で格納されるときの第2の署名付きコード片12の内部構造と等しいことを保証する。

【0024】第1の差分コード4が入出力ユニット6で受信された後、これが第2のソフトウェア・アーカイブ生成器7へ転送される。ここで、第1のコード片11が最初にそのソフトウェア・コンポーネント9に分離され、これらが次いで、生成命令8および第1の差分コード4に含まれた命令に従って、コンポーネント併合ユニット21によって併合される。この結果は、その署名Sig12を有する第2の署名付きコード片12である。よって、公開鍵15の使用下での暗号化検証ユニットによる後続の検証は、肯定的な検証結果19の結果にならなければならない。

【0025】この方法では、第2のコード片12全体ではなく、差分コード4のみを伝送すればよい。大きいコード片の場合、第2のコード片12と差分コード4の間のサイズにおける差は、たとえば電子伝送用の転送コストを考えると、非常に大きく莫大な影響となる可能性がある。インターネットのような電子ネットワークを介した小さいコード片の伝送は、大きいサイズのコード片の伝送と比較して、要する時間がより少なく、より安価で、割り込みおよびデータ損失の傾向が少ない。この伝送は、電子ネットワークを介して行う必要はなく、印刷された形式、プリンタおよびスキャナ、メールシステムまたは手による搬送において、あるいは光ネットワークを介して、あるいは磁気的に格納されたデータとしてなど、いかなる人または機械可読形式において、いかなる適切な媒体を介して行うこともできる。差分コード4は、更新または差分ファイルとも呼ばれる。

【0026】第2のソフトウェア・アーカイブ生成器7は、更新4を処理することができ、すなわち、新しい更新4を、すでにユーザ20側にインストールされた既存の第1の署名付きコード片11に併合する。この利点は、更新4が、クライアント側の第2のソフトウェア・アーカイブ生成器7によって、更新中のソフトウェア11に署名するときにサーバ側に存在するものと厳密に同じコード・コンテナ構造を再作成するために厳守される、厳密な順序の内容を含むという事実にある。これにより、署名および内容に関する限り、クライアント側で、更新されたソフトウェア12をそのまま正しく再作成することができる。

【0027】第2の実施形態

上記で説明した例を拡張して、ユーザ20が、第1の署名付きコード片11から第3の署名付きコード片13に達することを望むと仮定する。この更新では、原則として2つのステップが必要であり、すなわち、第1の署名付きコード片11から第2の署名付きコード片12に達

するための、第1の差分コード4を使用する第1の更新、および、第2の署名付きコード片12から第3の署名付きコード片13に達するための、第2の差分コード5を使用する第2の更新である。このためのいくつかの可能性が存在する。

【0028】a) 差分コードが、厳密に、第1の署名付きコード片11から第3の署名付きコード片13への移行のために決定される。差分コード生成器10は、これらの2つの署名付きコード片11、13を比較することによって、これを行うことができる。

【0029】b) ユーザ20が2つの差分コード4、5を受信し、一方の更新が、第2のソフトウェア・アーカイブ生成器7において他方の更新の後に実行される。しかし、これはあまり的確でないと見なされ、ユーザ20に負担を与える、ユーザが、より不体裁でおそらくより遅い2つの更新に対処しなければならない。第1の更新における動作は、後の更新がこれらを取り消すので冗長的になっており、差分コード4、5がネットワークを介して伝送された場合に時間および帯域幅が無駄になる。この状況は、全体のアップグレードを実行するために必要な中間更新の数が増加するにつれて悪化する。

【0030】c) 2つの差分コード4、5が全体の差分コードに結合され、次いでこれがユーザ20に伝送される。

【0031】差分コード4、5は、ユーザ20の要求時に決定される必要はないが、その前のいかなる時点でも所定にすることもでき、たとえばデータベースにも格納することができる。コード修正要求への応答を、もちろん、差分コード4、5が事前生成されて以前に格納されたときよりも速くすることができる。しかし、非常に多数のn個の異なるバージョンVx.xの場合、この方法では、あらゆる可能な更新の組み合わせが事前生成されて格納されている場合、非常に多数の更新差分コード4、5、すなわち理論上はn*(n-1)/2個もの異なる差分コード4、5となる。簡素化されたスキームは、差分コード4を可能な組み合わせの部分集合から事前生成するのみとなる。一連の後続の更新の場合、この集合は、各バージョンVx.0からその後続のバージョンV(x+1).0までの差分コード4、5を含むことができる。これにより、差分コード4、5の連鎖が事前生成され、これがn-1個の異なる差分コード4、5のみを含む。いくつかの更新のステップを包含する更新の要求、すなわち差分コード4、5が到着する場合、対応する差分コード4、5を、上記のc)で示したように結合することができる。現実的な場合、このような更新要求の確率は、典型的には、1つのバージョンからその後続のバージョンへの更新要求の確率よりもはるかに低い。

【0032】ユーザ20とソフトウェア・プロバイダ25の間の通信を、認証スキームを使用して行うことがで

きる。2つの関係者、すなわち、ユーザ20およびソフトウェア・プロバイダ25は、それにより、相手方が信用できるかどうか、真の相手方であるように装っている別の相手方ではないかどうかを決定することができる。この認証処理が通信を容易にし、更新の後に検証で不正な更新の試みを認識する結果となる状況を回避する。

【0033】認証スキームは、以下のように統合することができます。

【0034】ソフトウェア・プロバイダ側のサーバ25が、クライアント認証を必要とするSSLプロトコルなど、認証プロトコルを動作中である。

【0035】サーバ25が、署名付きコード片11、12、13のリポジトリを維持し、これはソフトウェア・ファイルとも呼ばれ、先にユーザ20、すなわち顧客に渡されたすべての可能なバージョンを記述するバージョン番号によって索引付けされる。サーバ25は、所与のソフトウェア・ファイルを別のソフトウェア・ファイル、好ましくは最新のソフトウェア・バージョンに変換するために必要な、すべてのデータおよびすべての署名関連情報を含む署名付き差分ファイル4も、格納しておくことができる。

【0036】クライアント側で、SSLクライアント・コードなど、クライアント・コードを動作し、これを使用して上記サーバ25への安全な接続を確立することができる。

【0037】クライアント側の認証ソフトウェアがさらに、元のソフトウェア配布に含まれた暗号化証明書および鍵を、たとえば、元のソフトウェア・ファイル・コードを搬送する製品CDに付属している個別化されたディスク上に含み、これを使用して、ソフトウェア配布サーバ25によりクライアント認証プロトコルを実行することができる。このように、クライアント20は、通信するサーバ25が本物のサーバ25であると確かめることができ、サーバ25には、通信するクライアント20にソフトウェア更新4の資格があることが分かる。

【0038】ソフトウェア更新4が、この安全な接続を介してクライアント20へ伝送される。

【0039】クライアント20が、更新の内容を検査する。すなわち、詳細には、ソフトウェア更新4、5に含まれたバージョン情報が、それが適切であるか、場合によってはクライアント20上に存在する現在のバージョンに等しいかどうかについて検査される。

【0040】次いで、クライアント側のソフトウェアが必要なすべてのデータを抽出し、これを、すでにクライアント20に存在する既存のソフトウェア11に結合する。詳細には、すべての署名付きコード・コンテナの内容の順序が適切にリストアされ、署名が、サーバ側で最初に作成されたそれらの機能性および正当性を達成するようになる。

【0041】最後のステップで、サーバ25から受信さ

れた更新ファイル4、5に含まれたすべての署名Sig11、Sig12、Sig13が、適切なファイル、すなわち、たとえばNetscape JARまたはMicrosoft CABファイルなど、署名付きコード・コンテナに適用される。

【0042】上で概説したこの手法を、継続された更新を必要とし、たとえばMicrosoft CABまたはNetscape JARなど、上述のファイル・フォーマットの1つにより署名される、いかなる形式のデータにも適用することができる。

【0043】上記で説明したシナリオを、非コネクション型シナリオで増すか、あるいは置き換えることができ、この場合は差分ファイル自体が暗号化により保護される。これは、ソフトウェア配布サーバ25へのオンライン接続が確立できないか、あるいはそれが望ましくない場合、意味をなす。この場合、サーバ側のソフトウェア配布機能が、暗号化署名Sig11、Sig12、Sig13を差分ファイル4、5に適用し、これがクライアント側の更新ソフトウェアによって、上で概説したオンラインSSL接続確立におけるセキュア・セッション確立に相当する方法において検査される。この手法に適切なファイル・フォーマットは、内部で処理することができるものと同じフォーマットであり、すなわち、たとえば、Netscape/SunのJARまたはMicrosoft CABである。

【0044】認証がなくとも、署名Sig11、Sig12、Sig13が、ある程度のセキュリティを提供し、これは更新処理によって損なわれない。検証はすなわち、いかなる理由であれ、更新された第2のコード片12の一部である署名Sig11、Sig12、Sig13が正しくなかった場合、すなわち、公開鍵15を使用することによって計算される、予想された署名Sig11、Sig12、Sig13と合致しなかった場合、否定的な結果を出す。このような場合、合致しなかった署名Sig11、Sig12、Sig13が、何かがうまく行かなかつたこと、および、更新が信用されないとを信号で通知する。よって、ユーザ20が、更新された第2のコード片12を使用しないように決定することができ、これは、ある無許可の修正がセキュリティ上の問題を導入した可能性があり、これがユーザ20を害する可能性があるからである。ユーザ20が第1の署名付きコード片11のバックアップ・コピーを保持していた場合、更新されたバージョン12を削除することができ、なお最初の第1の署名付きコード片11を使用することができる。

【0045】説明した実施形態は、部分的に、ならびに全部を組み合わせることができる。理解のために、更新に言及するとき、本発明はソフトウェアの更新に制限されるものではなく、第1の署名付きコード片11から第2、第3のコード片12、13などへのステップを実行

するように適用することができるが、第1の署名付きコード片11のために作成された署名の利点および正当性を維持することに留意されたい。ソフトウェア・アーカイブ生成器2、7として、いかなるコード片のそれぞれのそのハードウェア・バージョンは、それが、それらを单一のソフトウェア製品またはコンピュータ・プログラムとして单一化するために、詳細には、それを顧客または他のエンティティに送達するために、ソフトウェア・コンポーネントと一緒にまとめる機能を実行することを意味する。

【0046】本発明を、ハードウェア、ソフトウェア、またはこれらの組み合わせにおいて実現できることは、当業者には明らかである。これは、集中方式においてある単一のコンピュータ・システム上で、あるいは、分散方式において実施することもでき、この場合は、異なる要素がいくつかの相互接続されたコンピュータまたはコンピュータ・システムに渡って広がっており、それにより、いかなる種類のコンピュータ・システムも、あるいは本明細書で説明した方法を実行するように適合された他の装置が適する。典型的なハードウェアおよびソフトウェアの組み合わせを、ロードかつ実行中に、本明細書で説明した方法を実行するようにコンピュータ・システムを制御する、コンピュータ・プログラムを有する汎用コンピュータ・システムにすることができる。本発明をコンピュータ・プログラム製品に埋め込むこともでき、これは、本明細書で説明した方法の実施を可能にするすべての機能を含み、コンピュータ・システムにロードされたときにこれらの方針を実行することができる。

【0047】この文脈におけるコンピュータ・プログラム手段またはコンピュータ・プログラムは、情報処理機能を有するシステムに、特定の機能を、直接あるいは以下のa)別の言語、コードまたは表記法への変換、b)異なる材料形式における複製のいずれかあるいは両方の後に実行せざるよう意図された命令セットの、いかなる言語、コードまたは表記法における、いかなる表現をも意味する。

【0048】まとめとして、本発明の構成に関して以下の事項を開示する。

【0049】(1) ソフトウェア取得エンティティ(20)が既存の第1の署名付きコード片(11)から第2の署名付きコード片(12、13)に達することができるようとする、ソフトウェア・プロバイダ(25)のための方法であって、両方のコード片(11、12、13)は、第1のソフトウェア・アーカイブ生成器(2)を使用することによって、生成命令(8)の使用下で生成されており、前記第1の署名付きコード片(11)から前記第2の署名付きコード片(12、13)に達するために必要なステップを含む差分コード(4、5)を前記ソフトウェア取得エンティティ(20)に提供するステップを含み、この差分コード(4、5)は、前記ソフ

トウェア取得エンティティ(20)側で、第2のソフトウェア・アーカイブ生成器(7)によって前記第1の署名付きコード片(11)と結合されて前記第2の署名付きコード片(12、13)を生成するように使用可能であり、それにより、前記第1のソフトウェア・アーカイブ生成器(2)によって両方のコード片(11、12、13)の前記生成のために使用されたそれらの生成命令(8)が前記第2のソフトウェア・アーカイブ生成器(7)に供給される方法。

(2) 前記生成命令(8)が、前記ソフトウェア・プロバイダ(25)によって、好ましくは前記第2のソフトウェア・アーカイブ生成器(7)と共に前記ソフトウェア取得エンティティ(20)に提供される、上記(1)に記載の方法。

(3) 前記コード片(11、12、13)が、秘密鍵(14)を使用して署名される、上記(1)または(2)に記載の方法。

(4) 前記署名付きコード片(11、12、13)が、前記ソフトウェア・プロバイダ(25)側の記憶装置(1)に格納される、上記(1)ないし(3)のいずれか一項に記載の方法。

(5) 前記差分コード(4、5)が、好ましくは前記第1のソフトウェア・アーカイブ生成器(2)によって作成され、前記第1のソフトウェア・アーカイブ生成器(2)が前記第2の署名付きコード片(12、13)を生成する、上記(1)ないし(4)のいずれか一項に記載の方法。

(6) 3つ以上のコード片(11、12、13)が格納される場合、前記差分コード(4、5)が前記コード片(11、12、13)の部分集合の間でのみ生成される、上記(1)ないし(5)のいずれか一項に記載の方法。

(7) 前記第1のコード片(11)から前記第2のコード片(13)に達するために、いくつかの差分コード(4、5)が必要とされ、これらの差分コード(4、5)が単一の差分コードに併合されて前記ソフトウェア取得エンティティ(20)に提供される、上記(6)に記載の方法。

(8) 前記第1および第2のコード片(11、12、13)が、前記ソフトウェア・プロバイダ(25)側で、前記ソフトウェア取得エンティティ(20)から受信された要求(16)から対応する識別子を導出することによって識別される、上記(1)ないし(7)のいずれか一項に記載の方法。

(9) 既存の第1の署名付きコード片(11)から第2の署名付きコード片(12、13)に達するソフトウェア取得エンティティ(20)のための方法であって、両方のコード片(11、12、13)は、ソフトウェア・プロバイダ(25)側で、第1のソフトウェア・アーカイブ生成器(2)を使用することによって、生成命令

(8)の使用下で生成されており、前記第1の署名付きコード片(11)から前記第2の署名付きコード片(12、13)に達するために必要なステップを含む差分コード(4、5)の送達のために、コード修正要求(16)を前記ソフトウェア・プロバイダ(25)に送信するステップと、前記差分コード(4、5)を受信するステップと、第2のソフトウェア・アーカイブ生成器(7)を使用することによって前記差分コード(4、5)を前記第1の署名付きコード片(11)と結合し、それにより前記第2の署名付きコード片(12、13)を生成するステップとを含み、それにより、前記第1のソフトウェア・アーカイブ生成器(2)によって両方のコード片(11、12、13)の前記生成のために使用されたそれらの生成命令(8)が前記第2のソフトウェア・アーカイブ生成器(7)に供給される方法。

(10) 前記生成命令(8)が前記ソフトウェア・プロバイダ(25)から、好ましくは前記第2のソフトウェア・アーカイブ生成器(7)と共に受信される、上記(9)に記載の方法。

(11) 前記コード片(11、12、13)が、秘密鍵(14)を使用することによって署名され、署名(Sig11、Sig12、Sig13)が、対応する公開鍵(15)を使用することによって検証可能である、上記(9)または(10)に記載の方法。

(12) 前記第1および第2のコード片(11、12、13)が、前記コード修正要求(16)において対応する識別子を与えることによって前記ソフトウェア取得エンティティ(20)によって識別される、上記(9)ないし(11)のいずれか一項に記載の方法。

(13) 上記(1)ないし(12)のいずれか一項に記載の方法を実行するためのプログラム・コード手段を含む、コンピュータ・プログラム製品。

(14) コンピュータ可読媒体上に格納された前記プログラム・コード手段を含む、上記(13)に記載のコンピュータ・プログラム製品。

(15) ソフトウェア取得エンティティ(20)が、既存の第1の署名付きコード片(11)から第2の署名付きコード片(12、13)に達することができるようになるコード修正イネーブラであって、両方のコード片

(11、12、13)は、第1のソフトウェア・アーカイブ生成器(2)を使用することによって、生成命令(8)の使用下で生成されており、前記第1の署名付きコード片(11)から前記第2の署名付きコード片(12、13)に達するために必要なステップを含む差分コード(4、5)を生成するための差分コード生成器(10)を含み、この差分コード(4、5)は、前記ソフトウェア取得エンティティ(20)側で、第2のソフトウェア・アーカイブ生成器(7)によって前記第1の署名付きコード片(11)と結合されて前記第2の署名付きコード片(12、13)を生成するように使用可能であ

り、それにより、前記生成命令(8)が前記第2のソフトウェア・アーカイブ生成器(7)に供給され、さらに、前記ソフトウェア取得エンティティ(20)に前記差分コード(4、5)を提供するための出力ユニット(3)を含む、コード修正イネーブラ。

(16) 前記差分コード(4、5)の送達のためのコード修正要求(16)を前記ソフトウェア取得エンティティ(20)から受信するための入力ユニット(24)をさらに含む、上記(15)に記載のコード修正イネーブラ。

(17) 前記コード片(11、12、13)を生成命令(8)の使用下で生成するための第1のソフトウェア・アーカイブ生成器(2)をさらに含む、上記(15)または(16)に記載のコード修正イネーブラ。

(18) 既存の第1の署名付きコード片(11)から第2の署名付きコード片(12、13)に達するためのコード修正デバイスであって、両方のコード片(11、12、13)は、ソフトウェア・プロバイダ(25)側で、第1のソフトウェア・アーカイブ生成器(2)を使用することによって、生成命令(8)の使用下で生成されており、受信された差分コード(4、5)を前記第1の署名付きコード片(11)と結合し、それにより前記第2の署名付きコード片(12、13)を生成するための第2のソフトウェア・アーカイブ生成器(7)を含み、それにより、前記第1のソフトウェア・アーカイブ生成器(2)によって両方のコード片(11、12、13)の前記生成のために使用されたそれらの生成命令(8)が前記第2のソフトウェア・アーカイブ生成器(7)に供給される、コード修正デバイス。

(19) コード修正要求(16)を前記ソフトウェア・プロバイダ(25)に送信し、前記差分コード(4、5)を受信するための入出力ユニット(6)をさらに含む、上記(18)に記載のコード修正デバイス。

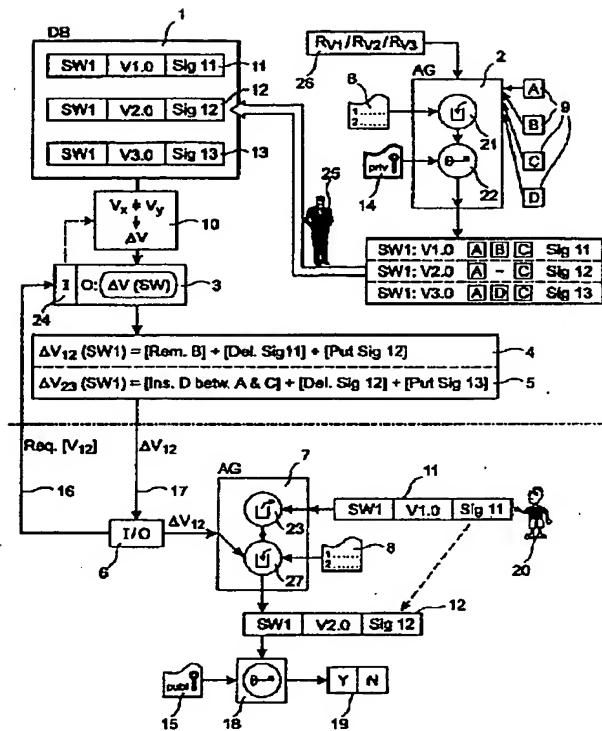
【図面の簡単な説明】

【図1】ユーザとソフトウェア・プロバイダの間のコードの交換において関係するステップおよびユニットの概略図である。

【符号の説明】

10	1 記憶装置 2 第1のソフトウェア・アーカイブ生成器 3 出力ユニット 4 差分コード 5 差分コード 7 第2のソフトウェア・アーカイブ生成器 8 生成命令 10 差分コード生成器 11 第1の署名付きコード片 12 第2の署名付きコード片 20 13 第2の署名付きコード片 14 秘密鍵 15 公開鍵 16 コード修正要求 20 ソフトウェア取得エンティティ 24 入力ユニット 25 ソフトウェア・プロバイダ S11 署名 S12 署名 S13 署名
----	---

【図1】



フロントページの続き

(51) Int.C1. ⁷	識別記号	F I	テーマコード(参考)
G 0 6 F 13/00	5 3 0	G 0 9 C 1/00	6 4 0 B
15/00	3 3 0	G 0 6 F 9/06	6 6 0 G
G 0 9 C 1/00	6 4 0		6 4 0 A

(72)発明者 ミヒヤエル・ベンチェ
スイス シー・エイチ8135 ラングナウ
アムアルビス シルワルト・ストラーセ
4

(72)発明者 ペーター・ブラー
スイス シー・エイチ8803 リュシュリコ
ン ミュエレストラーセ 39

(72)発明者 トーマス・エイリッヒ
スイス シー・エイチ8804 アウ ゾップ
フストラーセ 16

(72)発明者 フランク・ヘリング
スイス シー・エイチ8006 チューリッヒ
クルマンストラーセ 39

(72)発明者 トーマス・ヴァイゴルト
スイス シー・エイチ8134 アドリスヴィ
ル ブッテナウストラーセ 20

Fターム(参考) 5B017 AA03 BA07 CA15
5B076 AB10 BB06 FA00
5B082 CA04 GA11
5B085 AE13 BG07
5J104 AA09 LA03 NA02